REMARKS

In the Office Action of August 24, 2004, daims 1-10 directed to a system, method and software for providing persistence of complex data objects and their data relationships. The Examiner has objected to drawings, objected to the specification, objected to claims, rejected claims, or indicated allowable subject matter as follows:

1. The drawings were objected to under 37 CFR § 1.83, as needing to include name labels for and the relationship of features in the drawings for clarity.

2. Claims 4-6 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite because of the term "generic ebj stateful session bean" as not having support in the specification.

3. Claims 1- 3 and 7 were rejected under 35 U.S.C. § 102(a) as being anticipated by a U.S. Patent 6,314,434 issued to Shigemi et al.et al.

4. Claims 4-6 and 8-10 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Number 6,314,434 issued to Shigemi et al.et al. ("Shigemi") and in view of Published U.S. Patent Application (U.S. 2003/0195997) to Ibert et al.

With the above amendment, claims 1-10 (all of the prior claims) have been canceled without prejudice, and new Claims 11-20 have been added that correspond to the amended subject matter of cancelled claims 1-10. Such amendments clarify that a computer system is being claimed in claims 1-3 that has a memory and is loaded with object programming software modules rather than a data storage system. The amendments also clarify that the software modules in the claims are object language software modules. Such amendments are made to expedite the prosecution of this case, only. Applicant reserves the right to assert the deleted or canceled subject matter in this case, or in a continuing application. Reconsideration of this subject matter is respectfully requested in view of the traversal of the prior art rejection included herein. After entry of the above amendment, Claims 11-20 are active in the case. The above rejections are addressed in part by the present amendments and are otherwise traversed by the arguments that follow.

REPLACEMENT DRAWINGS

Replacement Figures 1 and 2 are appended to this response. The main labels in each drawing have been changed to indicate the Complex Data Object Graph (CDOG) and the Complex Data Object (CDOs) that make the CDOG. Applicant has complied with the Examiner's request merely to expedite the prosecution of this case. Applicant submits that the prior drawings were adequate formal drawings that provided clear numbering and descriptions of the numbered elements in the specification. All of the terms referred to in the drawing are clarified by the definition of terms that were part of the original application.

It is important for the Examiner to understand that an object within the meaning of this application is an object within the object programming language context that is interchangeable with the term "class" such as a Java Class. While a java class may enclose or have the ability to wrapper data to provide a "data object" in an object programming application environment, this is entirely different form a chunk of data located within a relational data store storage device. Data objects and complex data objects in an object programming context exist independent of the hard structure of the data storage device. Through a software mapping layer the Java Data Objects can retrieve data from the data store and may have entirely different relationships defined within as a object programming object and its relationship to other object programming objects from those defined within the mass storage device of the relational data store. The relationships between object programming objects (e.g, Java Classes), the data they contain to become defined as a "data object" and their relationships to other data objects are defined herein as a CDO. A graph showing the relationships between several CDO is defined as a

CDOG. The amended drawings clearly and succinctly illustrate these concepts of the present invention.

While a mass data storage facility, such as the structured data management system Shigemi et al.(cited by the Examiner), may store complicated as chunks and need to coordinate statements seeking to retrieve data from the data store, the Shigemi et al.description describes a data store, not an object programming application within a computer system as illustrated by the drawings and the present application description of CDO and CDOGs. Objects within the context of a relational data source and any graphical representation of the relational data store have distinctly different meanings from the use of the term "object" in the context of the drawings and this application. To better help the Examiner understand the amended drawings and its labels, some of the important definitions from the specification are set forth below for the convenience of the Examiner.

"Class" as referred to in this document in the context of computer software applications is a logic unit in a computer application or a computer software program where the application or program is based upon an objected oriented programming language (*e.g.*, Java). In practice, a class is a logical unit used as a logical template in an object oriented language from which to allocate new instances of objects.

"Object" as used in the context of this document is a general term referring to a logic unit in a computer application or a computer software program where the application or program is based upon an objected oriented programming language (*e.g.*, Java). The term "object" may ordinarily be used interchangeably with the term "class" as a template or as an instance depending on the context.

"Data object" as referred to in the context of this document represents the concept of the occurrence of an object that holds data within a specific computer application domain and is likely to have its contents stored in a persistent data source of a computer system (e.g., a database server, a binary file, a text file, or even in a combination of two or more of such a persistent data sources of a computer system). A data object may exist as an independent data object without any relationship to any other data object or it may have one or more relationships with itself or with one or more other data objects.

"Complex data object" (or "CDO") as used in the context of this document refers to the occurrence of a data object that has at least one or more relationships with itself, or at least one or more relationships with one or more other data object(s). In a given instance of a CDO at least one relationship is populated as a link, as defined below. A CDO may have a multiplicity of different relationships with itself or with one or more additional CDOs.

"Relationship" or "data relationship" as used in the context of a CDO refers to the type of logical combination that occurs between a data object with itself, or refers to the type of logical combination that occurs between a data object and at least one another data object. Among other references or descriptions, such a relationship is always referred to or partially described by a "relationship type". This term is used in an object oriented language context to reference or describe any expectations, actions and limitations possible between two or more data objects.

"Relationship type" in the context of this document is a label that specifies the possible multiple combinations that can occur between a CDO and itself or with at least one other CDO. The possible relationship type labels are 1-1 (one to one), 1-M (one to many) and M-M (many to many). A given CDO may be simultaneously related to more than one other CDO through several different types of relationship.

"Link" as used in this document with respect to a CDO identifies a particular occurrence of a relationship between a CDO and itself, between a CDO and another

CDO. The occurrence of at least one populated link results in an instance of the CDO.

Accordingly, the replacement sheets of drawings 1 and 2 are believed to overcome the Examiner's objections in view of the above explanation.

THE AMENDMENTS

New base claim 11 has been amended to clarify that system is a local or distributed computer system and to clarify that the computer system is loaded with at least one software module for maintaining transparent persistence. The software module claims have been clarified to indicate that an object programming language is utilized. Moreover the Examiner's attention is focused to the www.sun.com website and the resources for developers. The general EBJ specification and definitions are available there which clearly define the meaning of an ejb stateful session bean. This is further clarified by the present java code example of an ejb stateful session bean contained in Example 4, on pages 33-55 of the present specification. Each of claims has been amended to provide claims that are a computer system loaded for software that has the capability to accomplish the desired result as a result of an interface and the software that utilizes either the local or distributed computer system.

Moreover, regarding the software claims the software as defined has executable features that accomplish tangible results when loaded on a computer system. As recited in the claims, the computer program may be loaded on a local or distributed computer system and executed to utilize that system to create or maintain transparent persistence of complex data objects and their relationships as a CDOG model. The meaning of transparent persistence is defined in the present specification in that the data from complex data objects and their relationships within an application are being persisted without the application being in communication with the software that is persisting it as a CDOG..

THE 112(PARAGRAPH 2) REJECTION.

As clarified above, the phrase "generic ebj stateful session bean" is clear in view of the well-known meaning of the phrase "ebj stateful session bean" available to all Java programmers and developers from Sun Microsystems, Inc. (see their website www.sun.com) and look under developer resources for copious amounts of public documentation. The ebj session bean is generic as explained in the present specification in that it does not require modification to work with multiple applications, i.e., the java code for this session bean is generic. This is explained in the specification and illustrated by the generic session bean java code in Example 4, at pages 33-51 in the specification. Accordingly, this objection has been overcome. Withdrawal of this ground of rejection is respectfully requested.

THE 102(A) REJECTION UNDER SHIGEMI

Claims 1-3 and 7 were rejected under 35 U.S.C. § 102 (a) as being anticipated by U.S. Patent 6,314,434 issued to Shigemi et al.et al. As pointed out by the Examiner on page 3 of the office action, Shigemi et al.relates to a structured data management system mass storage device. Such is very different from the computer or distributed computer network system presently claimed that can provide transparent persistence of complex data objects as a CDOG.

The term "electronic data objects" referred to in Shigemi et al.is very different from the current definition of data objects, since data objects have a particular meaning within the context of an object programming language application from lay use of the term or from use of the term by relational data base mass storage experts.

A complex data object (CDO) within the present application is a object programming class "object" such as a Java Class that may wrapper data and then be referred to as a CDO or Java Data Object. Such object programming language terms have nothing to do with a relational data base tree structure or mass storage. Also, the claims have been clarified to provide that the system is a computer system with a working memory into which a software module is loaded that provides transparent persistence.

Therefore, Shigimi et al. does not even begin to teach or suggest the present invention, much less anticipate it. Accordingly, this ground of rejection is overcome.

THE 103 REJECTION UNDER SHIGEMI ET AL.ET AL. IN VIEW OF IBERT ET AL.

Claims 4-6 and 8-10 were rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Patent 6,314,434 issued to Shigemi et al.et al. in view of Published U.S. Patent Application (U.S. 2003/0195997) to Ibert et al. As pointed out by the Examiner on page 3 of the office action, Shigemi et al.et al. relates to a structured data management system mass storage device. This is very different from a computer or distributed computer network system presently claimed that can provide transparent persistence of complex data objects as a CDOG.

Moreover, the tree structure for data targets (referred to by Shigemi et al.et al. as "data objects") in Shigemi et al. does not correspond to the object programming data objects and the object model referred to in the present application. An object model of complex data objects within the Java object programming sense relates to Java Classes, for example, that may wrapper data and the logical relationships between different Java Class in their data, such as 1-1, 1-M (1-Many) and M-M. Logical relationships between complex data objects (CDO) in a Java Programming Application may be created or be change "on the fly" (dynamically) completely independently with respect to the hardwired nature of a mass storage devices, which was described by Shigemi et al. Such parameters and dynamic changes to the CDOs in the Java Software Application that may be running need to be persisted to a data store, particularly if someone else wants to use updated data or relationships. Therefore, the generic ejb stateful session bean (as shown in Example 4 and loaded in a computer system as described in the present specification) monitors changes to data objects and complex data objects of one or more applications that is running on the system and dynamically and transparently persists such changes as a CDO or CDOG without the need for an application to be aware of the persistence or the need for the application to request the system to provide it with such persistence as a CDO or CDOG.

The fact that a recorded set of logic is played as a stored macro in the Shigemi et al. mass storage device when a chunk of data (referred to by Shigemi et al. as a data object, meaning a data target…not a Java Class containing data) has nothing to do with transparent persistence as define in the present application.

Ibert et al. does not solve the deficiencies of Shigemi et al. There is no motivation in either Shigemi et al.et al.or Ibert et al. for combining the concepts from the two different technologies. Moreover, Ibert et al. does not teach transparent persistence of complex data objects or CDO or CDOGs. Instead, Ibert et al. relates to an adapter for a communication device. The fact that Ibert et al. uses the Java programming language to create the communication layer and perhaps even EJBs is irrelevant.

It is respectfully submitted that the Ibert et al. patent is inappropriate as a reference. In particular, the filing date of the Ibert et al. patent is October 21, 2002.

In contrast, one of the priority documents supporting the present invention, Provisional Application No. 60/316,075, was filed August 30, 2001. Support for the subject matter at issue is found on pages 20 –22 of the subject provisional application. Accordingly, it is urged that the combination of Shigemi et al. and Ibert et al. be withdrawn as Ground IV.

Moreover, since there is no ancillary reference cited that would teach the equivalence of the Shigemi et al. storage system to the presently claimed computer system, the present claims are also not obvious over the Shigemi et al. disclosure.

CONCLUSION

Accordingly, applicant respectfully submits that the above objections and rejections have been overcome, and should be withdrawn. In view of the amendments (including corrections of informalities) and remarks, the present application is believed to be in condition for allowance. Based upon the aforementioned comments and amendments, it is urged that the claims are in condition for allowance, as is the remainder of the subject patent application. Favorable reconsideration is respectfully requested.

Should the Examiner have any questions, comments, or suggestions, or should issues remain, he is respectfully requested to contact the undersigned by telephone for a prompt and satisfactory resolution.

Should any additional funds be needed, please draw from the deposit account no. 7154.

Respectfully submitted,
Lev Intellectual Property Consulting

Robert G. Lev
Registration No. 30,280

Date: February 24, 2005

4766 Michigan Boulevard
Youngstown, Ohio 44505
Telephone: (330) 759-1423
Facsimile: (330) 759-4865

I HEREBY CERTIFY THAT THIS CORRESPONDENCE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE AS EXPRESS MAIL IN AN ENVELOPE ADDRESSED TO: COMMISSIONER OF PATENTS, PO BOX 1450, ALEXANDRIA, VA 22313-1450, ON 02/22/05

LABEL NO. ER424903536US
NAME Amanda E. Ramos
SIGNATURE Amanda E. Ramos